THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 15 Article 1

August 2020

# A Two-Stage Model for Social Network Investigations in Digital Forensics

Anne David
*Cranfield University*, a.david@cranfield.ac.uk

Sarah Morris
*Cranfield University*, s.l.morris@cranfield.ac.uk

Gareth Appleby-Thomas
*Cranfield University*, g.thomas@cranfield.ac.uk

Follow this and additional works at: https://commons.erau.edu/jdfsl

 Part of the Computer Law Commons, Criminal Law Commons, Evidence Commons, Information Security Commons, and the Social Media Commons

## Recommended Citation

David, Anne; Morris, Sarah; and Appleby-Thomas, Gareth (2020) "A Two-Stage Model for Social Network Investigations in Digital Forensics," *Journal of Digital Forensics, Security and Law*: Vol. 15 , Article 1.
Available at: https://commons.erau.edu/jdfsl/vol15/iss2/1

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

(c)ADFSL

# A TWO-STAGE MODEL FOR SOCIAL NETWORK INVESTIGATIONS IN DIGITAL FORENSICS

Anne David [1], Sarah Morris [2] , Gareth Appleby-Thomas [3]
[123]Cranfield University
a.david@cranfield.ac.uk[1]
s.l.morris@cranfield.ac.uk[2]
g.thomas@cranfield.ac.uk[3]

**Keywords**: digital evidence, digital forensics, digital investigation, social networking, relationship attribution

## 1. INTRODUCTION

The proliferation of social networking sites has improved ways in which users engage and form relationships with people of similar interests, going beyond the days of email communications to the use of social networking applications to share messages, photos, and videos. It has also provided an opportunity for some users to perpetrate unlawful activities.

Social networking presents challenges to digital forensic investigations; for example, content posted may not always be written to permanent storage media. In addition, communication content can be altered or deleted after the fact. There is a need for digital forensic investigators to be able to recover such messages or other evidence that may be used to infer user activity and sufficiently attribute an action/actions to a user.

Evidence from social networking activity may be required in different types of criminal or corporate investigations. The type of evidence recovered helps the investigator obtain useful information that could:

- Guide the initial stages of an investigation, e.g., determining if a based on the evidence recovered; the suspect is worth investigating further
- Generate new leads which may lead to the:
    - Identification of other persons, places or items of interest
    - Identification of other potential sources of digital evidence to facilitate decision making

### 1.1 Contribution

The key contribution of this paper is the proposal of a two-stage model for evidence recovery and investigations involving social networking activities. It aims to help investigators prioritize digital evidence and maximize efficiency where resources are limited by focusing on extracting meaningful information from social networking artifacts. It is focused on the prompt identification and interpretation of associated artifacts and is aimed at enabling the analyst to quickly determine whether to expand or narrow the scope of an investigation.

### 1.2 Paper Structure

The rest of this paper is structured as follows: related work is discussed in Section 2. Section 3 proposes a two-stage model for social networking investigation in digital forensics. Section 4 discusses the experimental and analysis methodologies for this work. The research results and the implementation of the proposed model is presented in Section 5. Finally, the conclusions and potential future work stemming from this research are presented in Section 6.

## 2. LITERATURE REVIEW

This section provides a background on related work in digital forensics and social networking investigations. It discusses the requirements for the admissibility of digital evidence and the evidential

© 2020 JDFSL 

value of data generated as a result of user social networking activity.

### 2.1 Digital Evidence

Digital evidence can be described as any data that can be used to determine intent, culpability, how an event occurred, and the parties involved. It is useful in the investigation of a range of computer crimes and non-computer related crimes where evidence from a digital device may be used to link a suspect to an offense (Casey, Foundations of Digital Forensics, in Digital Evidence and Computer Crime. Forensic Science, Computers and the Internet, 2011).

Casey (2011) defines digital evidence as "data stored or transmitted using a computer that supports or refutes a theory on how an event occurred." Digital evidence is crucial in digital investigation and thus must be acquired in a forensically sound manner (McKemmish, 2008) to ensure that its admissibility in a court of law.

ACPO's (2012) definition of digital evidence encompasses a range of artifacts that can be found on digital devices, for example, system log files, application logs etc. Multiple devices with various artifacts, whilst ideal sources of digital evidence, present the challenge of "weeding out" information not directly relevant to the case. When time is of the essence, an investigator needs to be able to adequately identify devices that contain evidence pertinent to the case and use the intelligence obtained from those devices to progress the investigation.

Although the processing of digital evidence varies across jurisdictions, there are a few requirements digital evidence needs to meet before it is deemed admissible in court (Casey, 2005, Casey, 2002, Murr, 2007, Sommer, 1999):

- Evidence submitted must be **relevant** to the case.
- The evidence must be **reliable**
- The methods used to produce the evidence must be **repeatable** and should produce the same results when applied independently by a third party.
- The evidence must be **authentic** (genuine) and can be verified using hash values generated prior to and after imaging a device.
- The evidence must be **valid** and error-free. In exceptional circumstances where evidence acquisition from an active device is required, the process must be accurately documented, and any alteration accounted for.
- The evidence must be **trustworthy** and believable beyond a reasonable doubt.

In order to be admissible in court, evidence from social networking activity must satisfy these requirements.

### 2.2 Social Networking

Social networking has been defined as "the activity of sharing information and communicating with groups of people using the internet, especially through websites that are specially designed for this purpose" (Cambridge University Press, 2019). It enables users to connect with others and to form personal or business relationships.

In the context of digital evidence acquisition, social network activity provides a plethora of digital evidence to investigators. Artifacts from web browser history, cache, cookies etc. can be used to determine and infer a relationship between a user and a social network account or another user and may also be used to attribute an action to a user. This includes, but is not limited to, determining dates and times of access, usernames, session information etc.

In spite of the advantages presented by social networking applications (instant messaging, sharing personal events, microblogging, personal or corporate marketing, advertising etc.), it has also been known to present a means for a small minority of users to engage in disagreeable or criminal activities (Bello, 2013; Jonsson, 2011; Osborne, 2010; Richards, 2007; Rankin, 2012; House of Lords Select Committee on Communications, 2014; BBC News, Huge rise in social media "crimes", 2012; Moore, 2014; McGuire, 2019).

Investigating a user's (or suspect's) social network activity may be required for several reasons such as the collection of evidence to be used in court for the prosecution of an offender or for use in disciplinary actions taken against employees who abuse

© 2020 JDFSL 

corporate Acceptable Use Policy (AUP) (Taylor, Haggerty, Gresty, Almond, & Berry, T, 2014).

The Crown Prosecution Service (2018) provides guidance on how Prosecutors are to proceed to trial once they are satisfied with the evidence obtained during investigations involving social networking activity. However, there are no defined guidelines for digital forensic investigators with regard to prioritizing evidence collection and the management of artifacts related to social networking.

Taylor et al., (2014) suggests that although there are no specific guidelines for the forensic investigation of social networking applications, ACPO Guidelines can be used as a starting point for the investigation of offenses committed through or with a social networking application. It can thus be inferred that the lack of defined guidelines often results in such investigations being broadly categorized under 'web browser forensics' (Cusack & Son, 2012) due to the nature of access on a computer (while comparable, access through mobile devices is not considered here as it is considered outside the scope of this paper). However, it is suggested that focus on features specific to social networks, e.g., user IDs, profile IDs, etc. can also be used as a viable technique for evidence acquisition using methods tailored to web browser forensics.

Keyvanpour et al., (2014) present a three-phase framework for social network forensics; however, the specifics and potential location of artifacts of interest and techniques for recovery were not discussed. Oh et al., (2011) proposed an integrated method for the collection and analysis of web browser evidence where multiple web browsers have been used in the commission of an offense. This is based on the need to recover and utilize data created and stored on disk when a user accesses social networking sites using a web browser. It is important to note that due to the nature and flexibility of social networking applications, materials posted or shared can be later modified or deleted. In such situations, the service provider may be in a position to provide the evidence required to determine the author, when content was modified (or deleted) and reconstruct events.

Although at the time of writing there is a research gap in the use of social network artifacts as digital evidence (Taylor et al., 2014; Arshad et al., 2019; Das et al., 2016; Zainudin et al., 2011; Jang & Kwak, 2015; Huber, et al., 2011; Powell & Haynes, 2019), there are a number of reported cases where evidence from social networking activity has successfully led to prosecution (BBC News, 2010; Haroon & Carter, 2019; Bowcott et al., 2011; Press Association, 2014; Agency, 2015; Wood, 2018). In discussing evidence collection from social network activity, Arshad et al., (2019) grouped social network artifacts into four distinct classes, User, Activity, Network, and Content:

- User: consists of user data such as profile information, name, email address, phone numbers etc.
- Activity: consists of a timeline of user actions logged by the service provider on the server-side, e.g., dates and time of activity, location information, source of post, e.g., phone, tablet etc. These types of artifacts are created as a result of user actions on the social networking site; for example, when a user posts a comment about the service at a restaurant, the service provider tags the comment with the date and time, it's posted (see Figure 1). The location may also be included if geolocation is enabled. The user is unable to directly modify these types of artifacts.

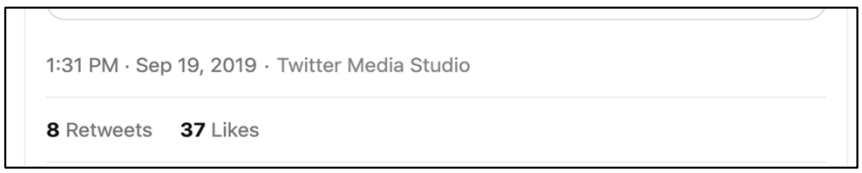



Figure 1 Activity time, date, source of tweet, number of retweets and likes illustrating server-side elements
Source: https://www.twitter.com

- Network: consists of personal social connections such as individuals or groups following or being followed by a user.
- Content: consists of materials published by a user such as photos, videos, tweets, retweets, shares etc.

Artifacts from each group can independently be used to infer user activity and, when combined, can be used to corroborate other related artifacts found on disk.

It is important to note that there is currently a knowledge gap with regard to formalizing the acquisition and analysis of social network artifacts. Some of the existing (traditional) digital forensic tools are not wholly designed for social networking investigations, and the capability for targeted searches or the ability of the tool to interpret and present the evidence in a human-readable format may be limited (Cusack & Son, 2012). For example, artifacts like Windows Registry Hives, Event logs, SQLite databases may need to be extracted and analyzed with a third-party tool. The objective of this paper is consequently to propose an approach that can be applied as a formal technique for social network investigations

### 2.3 Extracting Features from Social Networking Artifacts

Due to the proliferation of devices and the volume of data investigators need to process, it is often crucial to quickly identify the content of interest prior to a detailed analysis of a seized device. Feature extraction is an approach that enables investigators to process vast quantities of data in an efficient manner (see (Garfinkel, 2006; Garfinkel, 2013) for more on feature extraction).

A user's interaction with a digital device (computer) is a two-way process aptly explained by Locard's Exchange Principle (Chisum & Turvey, 2000) which states that every contact leaves a trace (Locard 1934, pp. 7-8 as cited in (Chisum & Turvey, 2007) (pp. 23-24)). With regard to digital evidence, this principle can be adopted to explain the existence of artifacts created as a result of user activity. For example, creating a user account, installing an application such as a web browser, or visiting social networking sites, all leave traces that can be used to infer what a user has done.

In the context of this paper, feature identification (and extraction) is described as the process of identifying and extracting artifacts containing key information about a user's social networking activities. Features in this context can be extracted from the absolute path of a given URL or other related artifacts such as HTML or JSON data, using pattern matching methods such as Regular Expressions (RegEx). The reoccurrence of a given feature can thus be attributed to a user's repeated access to a resource on a social networking site (Garfinkel, 2006).

#### 2.3.1 Identify Features in URLs

Every website visited by a user has a URL which indicates where resources are located, and the protocol used in accessing those resources. RFC 1738 describes URL as a compact string representation for a resource available via the Internet (Berners-Lee et al., 1994). It also describes the URL syntax as being made up of the following components:

***<scheme>:<scheme-specific-part>***

The <scheme> part defines the protocol used e.g. ftp, http. The <scheme-specific-part> varies and is dependent on the protocol used.

For example, two typical protocols comprise of:

*ftp://user@host.domain/directory/filename*

*or*

*http://domain_name/path/query-searchpart-parameter-fragment*

An example of a HTTP URL with three parts is shown in Figure 2. The query or search part of a URL may also be complex, having several parameters, as seen in Figure 3.

Some URLs may also specify subdomains or port numbers.




Figure 2 An example showing the parts of a HTTP URL




Figure 3 Example HTTP URL with multiple query or searchpart parameters

URLs can be generated in several ways, for example, clicking on a link in an email or a web page; clicking a bookmark or a shortcut; typing an address in the browser address bar, or using the autocomplete feature in the browser as shown in Figure 4.

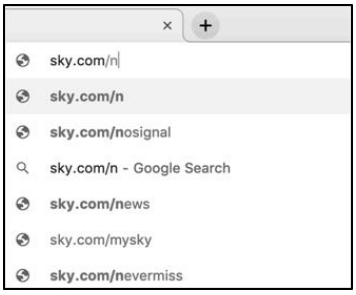



Figure 4 An example of the URL autocomplete/suggestion feature

URLs can provide much information about what a user has been doing online – from pages accessed to content searched for or shared. For this research, considering that URL structure is not browser dependent, identifying features in URLs involves decomposing the URL and decoding everything after the protocol part of the URL (i.e., <scheme-specific-part>) and in some instances, everything after the domain name part. This includes the path, parameters, and their values and fragments.

For investigations involving social network activities, finding and recovering actionable intelligence/evidence will help narrow the scope and refocus an investigation, thus maximizing the use of the investigator's time. Although there has been research in the forensic investigation of online activity and web browser artifacts, there is knowledge gap with regards to the deconstruction of individual URLs in relation to social networking activity. There is also limited research specifically focused on the modeling of social network forensic investigations processes and the extraction of features from deconstructed URLs.

The research described in this paper highlights the need for a concise model for the investigation and analysis of such social media related artifacts. It also highlights the importance of features in understanding user intent and the attribution of actions to a user.

## 3. PROPOSED MODEL

Building on the identified need for a new approach to forensic analysis of social media data, there are two key areas to consider - understanding user activity and the types of artifacts that this can generate, as well as the links between the two.

### 3.1 Defining "User Activity"

In general, there are three basic ways a user interacts with a computer:

1. Create/modify: This includes the creation of new files e.g., documents, modifying existing files, installing applications, file upload, or download.

2. Read/access: This includes accessing files with read, write, execute permissions. This type of interaction includes actions like launching a web browser (execute); generating an entry in the browser history (write), opening a folder or file (read).
3. Delete/remove files: This includes deliberate or unintentional removal of a file or application, e.g., uninstall, delete.

In the context of this paper, these activities can be broadly classified as:

- Pre-browser activities:
  - i.e., power on a device (computer), login, launch the web browser.
- Browser activities:
  - i.e., type in a URL in the address bar; click on a website shortcut; select autocomplete URL suggestion in the browser. This also includes accessing resources on a website, e.g., posts, uploads, downloads; such actions are recorded in the browser history, cache, etc.
- Post-browser activities:
  - i.e., close the web browser, some of the contents in memory are then written to disk, some are not recorded; power off the device.

At each point during any form of interaction with a device (pre-browser, browser, and post-browser), artifacts are created, allowing inference of what occurred. In any investigation where there is more than one user account on the device, to prove an action was initiated by a user, it is important to identify and highlight artifacts (including corroborating evidence) that determine whether the user being investigated was at the keyboard of the device in question within the timeframe of an incident.

### 3.2 Artifact Types

Artifacts found on a digital device can be broadly categorized into system generated and user generated artifacts (Mabuto & Venter, 2012):

- System generated: these are artifacts created by the operating system (OS) or an application on the computer without direct user action. System generated artifacts are created when core OS functions are performed by the OS or when a user performs core OS tasks. These artifacts can also be described as context artifacts. Examples include Event logs (user login/logout events); setup or configuration entries (created when a user enables/disables a function, installs an application, or when device drivers are installed).
- User generated: these are artifacts created as a result of a user's direct interaction with an application. For example, installing a web browser, accessing a website, creating a local document, downloading a file, deleting files, or uninstalling an application.

#### 3.2.1 Sources of Determination

There are several ways an investigator can build a picture or reconstruct an event and attribute actions to a user based on the artifacts recovered. These include but are not limited to using:

- Local Files: Created, modified, accessed dates and times can be determined when files resident on the device are analyzed.
- URLs: By deconstructing the URL, it is possible to determine what sites have been visited and what the user had typed in the address bar or search box, e.g., *search?q=statlerwaldorf&src=typd*
- System setup or configuration logs: This infers when an application was installed; a file was created, number of times run or accessed, last time an application was run, or a file was accessed. It may also contain the path to Event logs.
- System Event logs: Logging is a way for the OS to record information about system activities that occur. This includes date and time of the event; hostname/computer name of device involved; username of who was logged in to the machine when the event occurred; the program that triggered the event etc. In Windows, an identification number is assigned to each event type (Ultimate IT Security, 2014).

© 2020 JDFSL 

Attribution in the context of this paper requires an approach that links a user (or user account) to the web activity being investigated. The two-stage model proposed in this paper can be used to achieve this. This model is intended to produce case-specific and general artifacts in a social networking investigation:

i. Case-specific artifacts: these can be described as artifacts that are indicative of an action/activity of interest. For example, this feature "A" from the URL "Y" infers that the user clicked on the "X" tab on Twitter.
ii. General artifacts: these can be described as artifacts that provide an explanation of how an action occurred. These may also include artifacts expected to occur as a result of a user's activity on a social networking site. For example, clicking on a tab or link on a web page generates a new URL; however, HTTP headers or the JSON artifacts for the browser will indicate if a link was clicked as well as if there is a referrer URL. This is discussed further in Section 5.1.2.

### 3.3 Two-Stage Model

Details of the proposed model are presented in Figure 5.

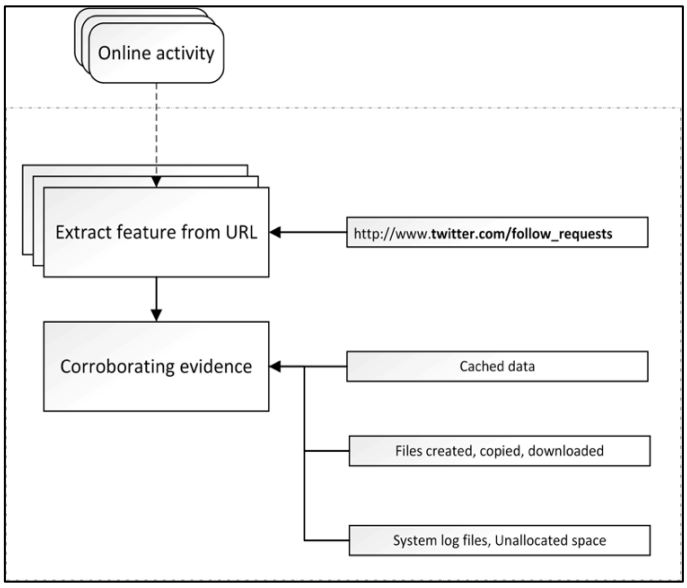



Figure 5 A schematic illustration of the proposed two-stage model for the investigations of social network activity

**Stage 1: URL feature extraction**

The first stage of this model involves the identification and recovery of URLs from the disk and the extraction of features from the URLs. The URL, in this instance, is the main/core source of features for online activity. For example, the social network site visited or the actions performed by the user (search, follow). It is important to note that URLs are not platform dependent, so this approach can be applied to any platform (i.e., OS or browser).

Features are extracted using a combination of RegEx and the sqlite3 module in Python. Artifacts recovered are stored in CSV files containing the dates and times of activity, the full URL, and extracted feature(s), which can be used to infer user activity or allude to the user's intent.

**Stage 2: Corroborating evidence**

Corroborating artifacts validate each piece of evidence found during an investigation. In the context of this paper, corroborating artifacts provide both confirmation and supplementary

information about the artifacts recovered during the URL feature extraction stage.

This stage of the model involves the identification and recovery of artifacts that validate what a URL feature indicates. These types of artifacts provide context to the features extracted from a URL. For example, the HTTP header information in the cache may show a URL that contains "*/settings/account*" was created as a result of clicking on the "account" link in the page "settings" causing the browser to respond by updating the URL and rendering the requested content. In addition to information derived from HTTP headers found in the browser cache (or unallocated space), metadata from the web page HTML could be useful in understanding the user's interaction with the social networking site.

This stage also involves the recovery of core OS artifacts that back up what has been inferred of the user's activity. For example, downloaded files associated with the recovered URLs; a local copy of uploaded data (associated with a "*www.domain_name/upload*" URL); artifacts indicating that a downloaded application was installed and run "*X*" number of times including the physical path of the application; artifacts verifying application paths.

The proposed model is useful for both the recovery of actionable intelligence and for focusing and ensuring a structured investigation. Having a "URL feature extraction" stage takes the bulk of URL artifacts and extracts meaningful information from them. This is useful because the digital forensic investigator needs a clear understanding of the URL structure in order to extract useable information from it.

When artifacts from the "URL" stage have been extracted, corroborating (supplementary) artifacts are used to contextualize events and help digital evidence meet the requirement to be beyond a reasonable doubt.

This paper presents work that improves on existing research on the forensic investigation of social network activities. It provides context by identifying and highlighting the importance of artifacts that corroborate or supplement the extracted feature(s). This includes data in URLs that ordinarily may be missed due to the volume of information returned by conventional digital forensics tools, data from HTTP headers, and general browser artifacts.

## 4. RESEARCH METHODOLOGY

This section presents the data generation and analysis methodology for this paper.

### 4.1 Data Generation

The experiments for this paper were conducted in virtual environments running Windows 7 and Windows 10, respectively, with the Firefox browser installed via an executable in a network shared folder. Firefox version 61 was installed with rolling updates up to version 69.

The purpose of these experiments was to simulate real-life activity on a social networking site and to create a feasible model for investigating such activities. The need to ensure the repeatability of the experiments made it necessary to use a virtual environment.

During the experimental phase, Fiddler (Telerik, 2018) was used to capture HTTP requests in order to understand how individual parts of a URL can be deconstructed; what was sent to the webserver and the response returned to the client (web browser); what was cached eventually irrespective of "no-cache" options in the header etc.

Data generation for this paper involved creating a local user account in the Windows virtual machine (VM) and creating a user account on Twitter using Firefox. Normal user activity was simulated by conducting a variety of the activities listed below over the course of the experiment:

- Powering on the VM and log in
- Launch the web browser
- Login to the test user account on Twitter
- Searching for users to follow
- Sending tweets
- Reply and retweet

- Viewing and sending Direct Messages (DMs)
- Send follow requests to other test user accounts
- Viewing and accepting follow requests
- Updating the test user account privacy settings
- Continuous scrolling and viewing the test user account timeline

During the experiments, all activity was logged in a contemporaneous note as a means of verifying the user activity against the results found during the analysis.

### 4.2 Data Analysis

The analysis was conducted in a Windows 10 desktop environment using existing digital forensics tools. This was done to identify and understand artifacts of interest in a social networking investigation and to help with the implementation of the proposed model. The data analysis covered core OS artifacts and user generated artifacts, and a multi-tool analysis technique was employed. The following tools were used:

- General examination tool:
    - WinHex (X-Ways Software Technology AG, 2018): this was used to view the virtual disk contents and for the extraction of artifacts to be analyzed with third-party tools (listed below). It was also used for simultaneous keyword search. WinHex simultaneous search function allows a list of search terms (one per line) to be searched at once.

- Tools used for individual artifact analysis:
    - Registry Decoder (Case & Marziale): this was used to analyze the registry hives.
    - RegRipper (Carvey, 2018): this was used to validate the results from Registry Decoder.
    - DB Browser for SQLite (DB4S Project, n.d.): used for the analysis of user browser SQLite databases in the user's Firefox profile.
    - MZCacheView (NirSoft, 2018): used for the analysis of the Firefox cache artifacts.
    - FullEventLogView (NirSoft, FullEventLogView, 2018): used for the analysis of Windows event logs.
    - Prefetch Forensics (Woan, 2013): used for the analysis of prefetch artifacts.
    - Python 3 (Python.org, 2019): python scripts were used to convert the sessionstore files to JSON format and to extract features from extracted URLs.

## 5. RESULTS AND DISCUSSION

This section discusses the experimental results, following the proposed two-stage model approach. These are categorized into URL feature artifacts and corroborating artifacts.

### 5.1 Implementation

Artifacts of interest in this research include user and system generated artifacts related to user activity on Twitter. This includes URLs generated as a result of Twitter activity (login, searching for followers, viewing followers, tweeting etc.) and system generated artifacts that give context to a user's activity.

#### 5.1.1 URL Feature Artifacts

As discussed in previous sections, features from URLs can be used to give context to or infer user action. URL artifacts were recovered from the user's Firefox Profile using DB Browser for SQLite. It is important to differentiate between the History location and the Cache location as both folders share the same name. In this section, when the Firefox Profile folder is mentioned, it refers to the folder containing the browser history. The Cache is discussed in Section 5.1.2.3.

The Firefox profile can be found in: *%APPDATA%\Mozilla\Firefox\Profiles* where %APPDATA% is %SYSTEMROOT%\Users\<username>\AppData\ Roaming. The profile folder contains a subdirectory

with a *.default* extension e.g., *oeds8ys7.default*, which contains the SQLite database files (as shown in Figure 6), session information files, and other files and directories used by Firefox.

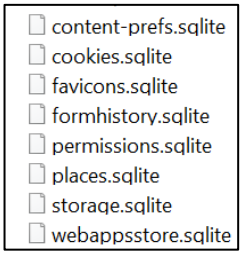



Figure 6 SQLite database files in the Firefox Profile folder

#### 5.1.1.1 History

The browser history is written to *places.sqlite* and was analyzed using DB Browser for SQLite. The query was focused on URLs indicative of accessing Twitter. The returned URLs include login, timeline, search, profile views, and follow/follower activities. Extracts of the results are shown in Figure 7 (although the experimental user is fictitious, certain parts of the URL have been modified to remove identifying information).

The extracts show standard Twitter activity URLs as displayed in the browser address bar during the experiments.

```
1  SELECT datetime(visit_date/1000000, "unixepoch"), url, title, rev_host, from_visit, visit_type, visit_count
2  FROM moz_places, moz_historyvisits
3  WHERE visit_date !=""
4  AND  moz_places.url LIKE "%twitter%"
5  ORDER BY visit_date
6
```

| | datetime(visit_date/1000000, "unixepoch") | url | title | rev_host | from_visit | visit_type | visit_count |
|---|---|---|---|---|---|---|---|
| 1 | 2018-12-01 18:18:12 | http://twitter.com/ | NULL | moc.rettiwt. | 0 | 1 | 1 |
| 2 | 2018-12-01 18:18:12 | https://twitter.com/ | Twitter | moc.rettiwt. | 0 | 1 | 6 |
| 3 | 2018-12-01 18:18:12 | https://twitter.com/login | Login on Twitter | moc.rettiwt. | 0 | 1 | 1 |
| 4 | 2018-12-01 18:18:12 | https://twitter.com/i/notifications | Twitter / Notifications | moc.rettiwt. | 0 | 1 | 1 |
| 5 | 2018-12-01 18:18:12 | https://twitter.com/settings/account | Twitter / Settings | moc.rettiwt. | 0 | 1 | 1 |
| 6 | 2018-12-01 18:18:12 | https://twitter.com/settings/safety | Twitter / Settings | moc.rettiwt. | 0 | 1 | 1 |
| 7 | 2018-12-01 18:18:12 | https://twitter.com/N[redacted]d/followers | People following O[redacted]s (@N[redacted]d) \| Twitter | moc.rettiwt. | 0 | 1 | 3 |
| 8 | 2018-12-01 18:18:12 | https://twitter.com/follower_requests | O[redacted]s (@N[redacted]d) \| Twitter | moc.rettiwt. | 0 | 1 | 1 |
| 9 | 2018-12-01 18:18:12 | https://twitter.com/S[redacted]3 | S[redacted]h (@S[redacted]3) \| Twitter | moc.rettiwt. | 0 | 1 | 1 |

```
Result: 240 rows returned in 96ms
At line 1:
SELECT datetime(visit_date/1000000, "unixepoch"), url, title, rev_host, from_visit, visit_type, visit_count
FROM moz_places, moz_historyvisits
WHERE visit_date !=""
AND  moz_places.url LIKE "%twitter%"
ORDER BY visit_date
```

Figure 7 URLs showing user activity (recovered from places.sqlite)

The results from DB Browser for SQLite were exported and saved in .csv format ready for URL deconstruction.

Using the contemporaneous notes, URLs were grouped based on user activity. Tables 1 - 3 show a profile/timeline view, clicking the followers link on the user's timeline and viewing follower requests, respectively.

Table 1 Timeline URL and tab title

| URL | Tab title |
|---|---|

| | |
|---|---|
| https://twitter.com/NxxxxxxxOxxx<br>*(User's home/timeline)* | Oxxx Nxxxxxxx (@Nxxxxxxxxxxd) \| Twitter<br>(*Tab title contains user's full name and (@user's handle)*) |

Table 2 URL generated when the "followers" link is clicked

| URL | Tab title |
|---|---|
| https://twitter.com/NxxxxxxxOxxx/followers<br>(*Created when "followers" is clicked*) | People following Oxxx Nxxxxxxx (@Nxxxxxxxxxxd)<br>(*Tab title*) |

Table 3 URL generated when the "followers" link is clicked to view follow requests

| URL | Tab title |
|---|---|
| https://twitter.com/NxxxxxxxOxxx/follower_requests<br>(*"pending follower requests" is clicked.)* | Oxxx Nxxxxxxx (@Nxxxxxxxxxxd) \| Twitter<br>(*Tab title*) |

During the experiments, the account security and privacy settings were updated. This includes making the account private etc. The user clicked on settings and was directed to the account page from where security and privacy settings can be modified. Tables 4 - 5 show URLs from this activity:

Table 4 URL indicative of account settings modification

| URL |
|---|
| https://twitter.com/settings/account<br>(*This URL takes the user to the 'Account page' from where the user account settings can be modified.)* |

Table 5 URLs indicative of privacy settings modification

| URL |
|---|
| https://twitter.com/settings/safety<br>(*When the user clicks on 'Privacy and safety' within settings, this URL is created. This page allows the user to set tweets as private, disable location tagging etc.*) |

To break down the URLs into manageable components, the CSV and re modules in Python 3 were used, as seen in the example code extract below. The features including the date and time of access were written to a CSV file using a version of the code illustrated below:

```
import csv
import re

# open the csv file
with open("history.csv") as hist:
   # read the csv file
   readCSV = csv.reader(hist, delimiter=',')
  for row in  csv file
     for item in  row
        if 'twitter' in item
          pattern = your_regex_pattern
          regex_search = re.compile(pattern, re.IGNORECASE)
         matches = regexp.findall(item)

        # write regex match to a csv file
       with open("matched_patterns.csv", 'a') as mp:
      mp.write("{0}, {1}".format(row[0], ','.join(matches)) + '\n')
```

The examples below are some of the saved feature matches; they are intended to help an analyst make sense of how an event happened. For example, the user accessed Twitter on date and time; then, the user navigated to this part of the page; the user searched for this other user etc.

When sorted by the date and time of activity, it can be used to recreate a probable timeline of activity.

```
2016-01-17 21H:09M:49S, twitter.com, MissPiggy?ref_src=twsrc%5Egoogle
2018-12-01 18H:18M:12S, twitter.com, login
2018-12-01 18H:18M:13S, twitter.com, i, notifications
2018-12-01 18H:21M:14S, twitter.com, settings, account
2018-12-01 18H:21M:14S, twitter.com, settings, safety
2018-12-01 18H:21M:52S, twitter.com, Nxxxxxxxxxd, followers
2018-12-01 18H:21M:52S, twitter.com, follower_requests
2018-12-01 18H:21M:52S, twitter.com, Sxxxxxxh3
2018-12-01 18H:21M:52S, twitter.com, Nxxxxxxxxxd
2019-11-08 11H:46M:15S, twitter.com, Pxxxxxxxxd
2019-11-11 15H:14M:45S, twitter.com, DxxxxxKxxxx, with_replies
2019-12-24 11H:53M:39S, twitter.com, jxxxxhxxxxx, status, 544385844081987584
```

© 20

### 5.1.2 Corroborating Artifacts

Corroborating artifacts, as discussed in Section 3.3 are artifacts that provide context to the URL artifacts recovered using the two-stage model proposed in this paper. This section discusses corroborating artifacts as they relate to the URL artifacts discussed in the preceding section.

#### 5.1.2.1 Other SQLite Database Files

Cookies.sqlite records cookies set during a browsing session. It provided corroborating information for the features that were recovered from URLs in Section 5.1.1.1. Figure 8 shows a cookie set for the path "settings/safety".

| baseDomain | name | value | host | path |
|---|---|---|---|---|
| twitter.com | dnt | 1 | .twitter.com | /settings/safety/ |
| twitter.com | fm | 0 | .twitter.com | / |

Figure 8 Cookies set in cookies.sqlite indicates the setting/safety page was accessed on twitter.com

The cookie information can be used in conjunction with other artifacts to determine and link the browsing session where the activity of interest occurred, as shown in Figure 9.

| fieldname | value | datetime(lastUsed/1000000, "unixepoch") |
|---|---|---|
| session[username_or_email] | o[redacted]8@gmail.com | 2018-12-01 18:35:01 |

Figure 9 Login credentials from formhistory.sqlite shows the last time the recorded email address was used

Formhistory.sqlite records entries made in form fields in Firefox. These are stored in *key:value* pairs where the fieldname is the key and the entry typed into a text field is the value. Information recovered shows the login username for the user's Twitter account and the last time the username was used.

Webappsstore.sqlite stores data for websites in *key:value* pairs. In this instance, the value of the *__typeahead__:userHash* key contained information on the user's Twitter account including that of over 500 other Twitter users. This amount of information can be overwhelming; however, using features such as *@user_handle*, *user_id*, *profile_id*, extracted from the URLs, it can be filtered down to a manageable size.

Artifacts from webappsstore such as the values of "*followed_by*" and "*following*" within "social_context" as seen in Figure 10, can be used to infer a user's social connection (relationship) to other users.

© 2020 JDFSL 

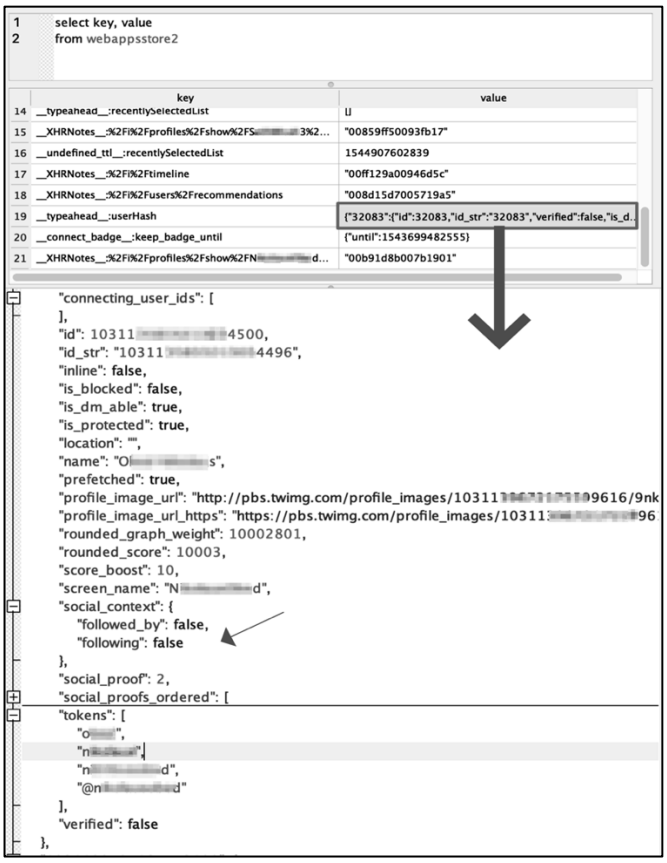



Figure 10 Extracts from webappsstore.sqlite contain JSON data that can be used to determine a user's follow/following status on Twitter

#### 5.1.2.2 Session Information

Artifacts supporting follow activity identified during the URL extraction stage were recovered from Firefox sessionstore. Sessionstore (as at the time of writing) is stored in a compressed file format (MOZLZ4/JSONLZ4) in the browser profile folder and is used by Firefox to manage the ability to restore currently open windows and tabs in the event of a crash or forced restart. It can also be used to open previous tabs on startup following a clean exit.

The following files are also used to store session data:

*%APPDATA%\Mozilla\Firefox\Profiles\<profile folder>\sessionstore-backups*

- previous.jsonlz4
- recovery.baklz4
- upgrade.jsonlz4-[datetimestamp] (session state before a browser version upgrade)

The information of interest in sessionstore includes URL, page title, referrer URL; time a tab was last accessed or closed; the time a window was last accessed or closed; session start/last updated time; cookies associated with the session.

The sessionstore file was decompressed from LZ4 to JSON format, using the lz4 module in Python 3. When analyzed, it provided information on the user session and corroborated the activities inferred by the URLs (Section 5.1.1), as shown in Figures 11 - 14.

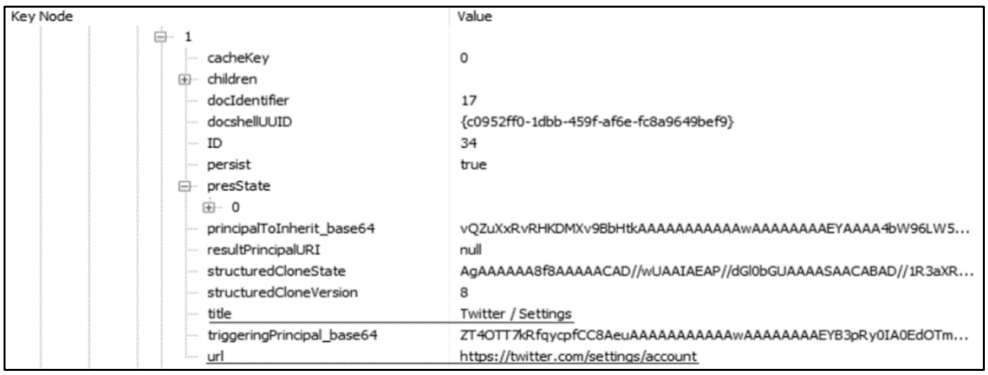



Figure 11 JSON (sessionstore) artifacts showing session where user account settings were accessed

JSON artifacts, such as the extracts shown in Figure 12, are useful in identifying the session a URL is a part of and any associated (referrer) URL.

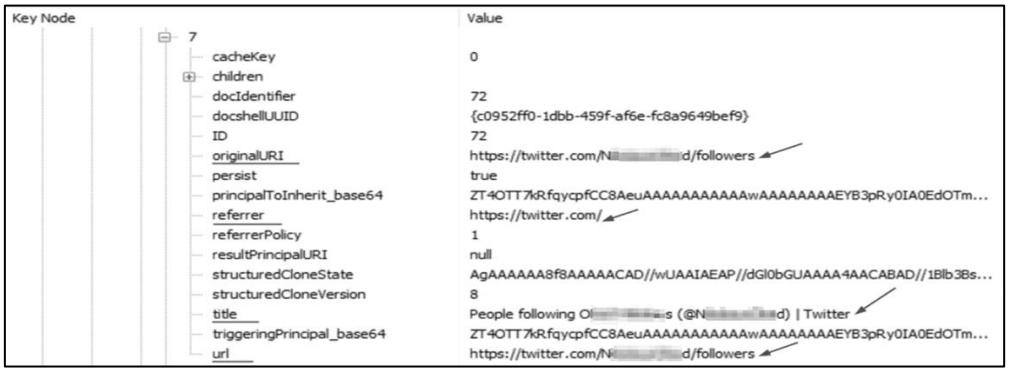



Figure 12 JSON (sessionstore) artifacts showing referrer URL through which the user reached "followers"

When the URL contains a username other than the logged in user, it indicates a visit to the other user's timeline/profile.

Figure 13 shows a visit to the profile (URL) and the tweet sent by the experimental user.




Figure 13 JSON (sessionstore) artifacts indicate the user visited another user's profile and sent a tweet

In order to establish timelines, it is important to view the session within which the activity of interest occurred as shown in Figure 14.

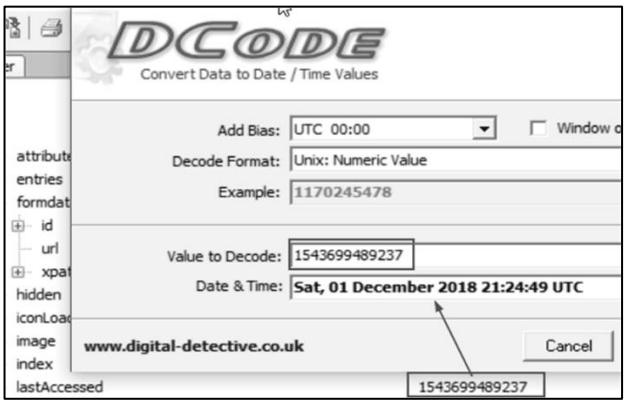



Figure 14 JSON (sessionstore) artifacts showing the last accessed date and time for the session

#### 5.1.2.3 Cache

The Firefox cache can be found in: %SYSTEMROOT%\Users\<username>\AppData\Local\Mozilla\Firefox\Profile. The profile directory contains a subdirectory with a *.default* extension and an identical name to that of the history profile folder (see Section 5.1.1).

During the analysis of the cache, artifacts recovered were indicative of a user's direct interaction with other users, for example, profile banners and profile photo URLs and images. These URLs validated the contents of webappsstore.sqlite and sessionstore and are shown below in Figures 15 and 16:




Figure 15 Extract of user profile data from webappsstore.sqlite




Figure 16 Extract of user profile image from the cache

Some of the *http://pbs.twimg.com* URLs were for profiles the user did not interact with. Just as with the webappsstore artifacts, the URLs can be grouped into a separate category after the features extracted in Section 5.1.1.1 have been used to filter out the URLs of interest, indicating social networking activity.

It is important to know that some of the URLs recovered from the cache were as a result of background processes on Twitter. For example, hashflags and hashtags. Hashflags can be described as custom emojis that accompany a hashtag (e.g., #StarWars ⚔) and are used by Twitter to promote events. For example:

```
https://abs.twimg.com/hashflags/Amazon_Holiday_2018/Amazon_Holiday_2018.png
https://abs.twimg.com/hashflags/WB_LegoMovie_Emmet/WB_LegoMovie_Emmet.png
```

© 2020 JDFSL 

Hashflag URLs may be cached even though they have not been viewed or used by the user. An example of a hashflag URL recovered from the cache:

```
7109699712   68 74 74 70 73 3A 2F 2F  61 62 73 2E 74 77 69 6D   https://abs.twim
7109699728   67 2E 63 6F 6D 2F 68 61  73 68 66 6C 61 67 73 2F   g.com/hashflags/
7109699744   57 42 5F 4C 65 67 6F 4D  6F 76 69 65 5F 45 6D 6D   WB_LegoMovie_Emm
7109699760   65 74 2F 57 42 5F 4C 65  67 6F 4D 6F 76 69 65 5F   et/WB_LegoMovie_
7109699776   45 6D 6D 65 74 2E 70 6E  67 00 E5 00 00 00 00 00   Emmet.png å
```

Hashtag URLs from the "Trends for you" frame on the user's Twitter homepage were also recovered. The user also did not interact with this part of Twitter or use hashtags during the experiment. Examples of hashtag URLs recovered from the cache are:

```
https://twitter.com/hashtag/Disney?src=hash
https://twitter.com/hashtag/RoaldDahl?src=hash
https://twitter.com/hashtag/StarTrek?src=hash
https://twitter.com/hashtag/WrathOfKhan?src=hash
```

An example of a hashtag URL recovered from the cache:

```
7096393600                        68 74 74 70 73 3A 2F 2F          https://
7096393616   74 77 69 74 74 65 72 2E  63 6F 6D 2F 68 61 73 68   twitter.com/hash
7096393632   74 61 67 2F 57 72 61 74  68 4F 66 4B 68 61 6E 3F   tag/WrathOfKhan?
7096393648   73 72 63 3D 68 61 73 68  00                       src=hash
```

Some of the recovered URLs were from suggested/promoted tweets advertised on the user's timeline. It is important to note that these ads/sponsored content can be found on disk even when a user hasn't clicked on them. In the context of this paper, they were a result of continuous scrolling on the user timeline. These tweets can be identified by the example features highlighted below:

https://twitter.com/**i/cards**/tfw/v1/1056932085485658113?**advertiser_name=NespressoUK&Ireland**&cardname=unified_card&**is_following_advertiser=false**&forward=true&impression_id=358a4182a9a96b66&edge=true&lang=en&card_height=271&**scribe_context={"client":"web","page":"home","section":"home","component":"tweet"}**&bearer_token=AAAAAAAAAAAAAAAAAAAAAPYXBAAAAAAACLXUNDekMxqa8h/40K4moUkGsoc=TYfbDKbT3jJPCEVnMYqilB28NHfOPqkca3qaAxGfsyKCs0wRbw#**xdm_e=https://twitter.com**&xdm_c=default4701&xdm_p=1

xdm is cross-domain messaging, and the expected value for *xdm_e* is the base URL of the host. As shown in the Google Analytics URL above, the base URL is *https:twitter.com*. To determine whether a user is following an advertiser, the value for "**is_following_advertiser**" would be "*true*".

Figure 17 shows an extract of HTTP Request and Response headers captured through Fiddler, which

was used to monitor Twitter traffic during the experiment. Fiddler was used to determine what was expected to be written to disk, what eventually was written to disk and what was not written to disk. The ad URLs captured by Fiddler corroborate the Twitter advertising URL recovered from the user's Firefox cache. They show that the background processes on Twitter can be written to disk.

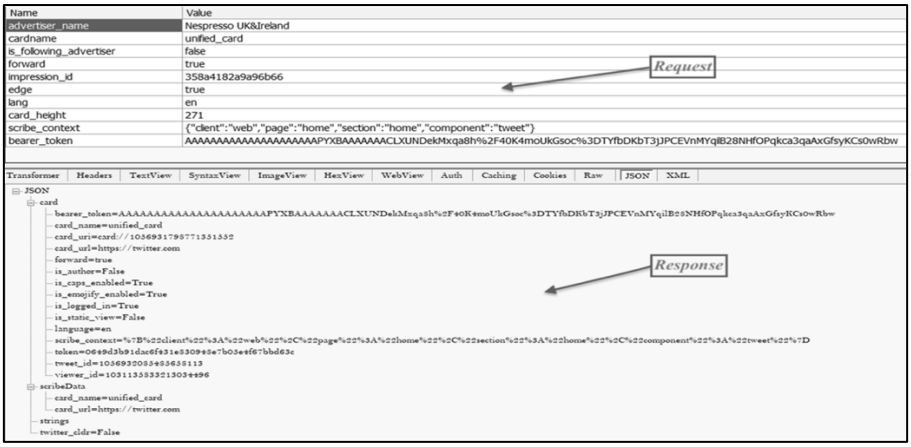



Figure 17 Extracts from Fiddler showing the HTTP Request and Response for the cached sponsored content

Due to the volume of cached data, and although they may not be used for user relationship attribution, it is important to identify and highlight social networking artifacts that are unrelated to direct user interaction. This would help an investigator/analyst focus on artifacts of immediate interest. For example, grouping sponsored content into different categories, separating them from normal user accounts.

#### 5.1.2.4 Registry and Event Logs

The artifacts presented in this section come from a range of sources (OS and user generated). They aim to answer questions such as the number of user accounts on the system, the logged in user; dates and times of activity; applications installed or accessed; paths to files and applications etc.

Where there are multiple user accounts on a device, it is important to identify registered accounts of interest, credentials, permissions, and dates and times of access. These can be identified and recovered by analyzing the SAM registry hive and reviewing Windows Event logs. The hives (Table 6) were extracted using WinHex and information about the user account, such as when the account was created, the username, the account type, application (browser) installation, and access were recovered.

Table 1 Registry hives extracted with WinHex

| **Hive** | **Location** |
|---|---|
| SAM<br>SECURITY<br>SOFTWARE<br>SYSTEM | %SYSTEMROOT%\System32\config |
| NTUSER.DAT | %SYSTEMROOT%\Users\<username> |
| UsrClass.dat | %SYSTEMROOT%\Users\<username>\AppData\Microsoft\Windows |

The SAM hive provided information on the user accounts, date and time created and last login time.

```
Username        : Oxxx Nxxxxxxx [1000]
Full Name       :
User Comment    :
Account Type    : Default Admin User
Account Created : Sun Sep 25 14:44:15 2016 Z
Name            :
Last Login Date : Sat Dec  1 18:12:26 2018 Z
Pwd Reset Date  : Sun Sep 25 14:44:15 2016 Z
Pwd Fail Date   : Never
Login Count     : 8
```

The last login date and time can be used to corroborate session information recovered sessionstore:

Extracts from the SYSTEM hive indicate the path of the Firefox installer in the network shared folder (Section 4.1) and an indication that Firefox was executed:

```
1533971040|REG|||M...                AppCompatCache                -
Z:\shared_installer_files\Firefox Setup 61.0.2.exe

1533670994|REG|||M...    [Program    Execution]    AppCompatCache    -
C:\Program Files\Mozilla Firefox\firefox.exe [Executed]
```

This type of information recovered from the Registry when cross referenced with Event logs (Figure 19) verifies user account information and may be used to link identities matching the username, e.g., registered social network credentials. It can also be used to attribute specific browser (social network activity) sessions to a user based on login/logoff times correlated using the "*LogonId*" as seen in Figures 18 and 19.

Other artifacts of interest were recovered from NTUSER.DAT, indicating when the browser (Firefox) was installed, the set default browser as shown in Table 7, where it was launched from and the number of times the browser was launched as shown in Table 8—for example, starting Firefox from the desktop or taskbar shortcut. This could help corroborate sessions and social network activity.

Table 2 Default browser setting from the Registry

| **Firefox is set as the default browser** |
|---|
| StartMenuInternet [Sat Aug 18 18:11:33 2018 (UTC)] |
| NOTE: default Internet Browser client key |
| (default) -> Firefox-308046B0AF4A39CB |

Table 3 Firefox executed once from Taskbar shortcut

| **Browser execution and Run count** | | |
|---|---|---|
| Datetime stamp | Path | Run count |
| Sat Dec 1 2018 18:19:15 | {9E3995AB-1F9C-4F13-B827-48B24B6C7174}\TaskBar\Firefox.lnk | 1 |

Event IDs 4720, 4624, and 4647 mean 'A user account was created', 'An account was successfully logged on' and 'User initiated logoff' respectively [40]. There are other description fields in the Windows event logs that can provide additional information (e.g., Account Security ID, Domain etc.), but they are out of scope for this paper. Figures 18, 19, and 20 show extracts from the event logs.

```
<EventID>4720</EventID>-
<TimeCreated SystemTime="2016-09-25T14:44:15.715800100Z"/>-
<Channel>Security</Channel>-
<EventData>-
<Data Name="TargetUserName">Oxxx Nxxxxxxx</Data>-
<Data Name="SamAccountName">Oxxx Nxxxxxxx</Data>-
<Data Name="LogonHours">%%1797</Data>-
</EventData>
```

Figure 18 Account creation dates, time, username (extract from Security Event log)

```
<EventID>4624</EventID>-
<TimeCreated SystemTime="2018-12-01T18:12:26.149464600Z"/>-
<Channel>Security</Channel>-
<EventData>-
<Data Name="TargetUserName">Oxxx Nxxxxxxx</Data>-
<Data Name="TargetLogonId">0x0000000000015a3c</Data>-
<Data Name="LogonType">2</Data>-
<Data Name="ProcessName">C:\Windows\System32\winlogon.exe</Data>-
<Data Name="IpAddress">127.0.0.1</Data>-
<Data Name="IpPort">0</Data>-
</EventData>-</Event>-
```

Figure 19 Login date, time, type and username (extract from Security Event log)

```
<EventID>4647</EventID>-
<TimeCreated SystemTime="2018-12-01T21:31:39.057611700Z"/>-
<Channel>Security</Channel>-
<EventData>-
<Data Name="TargetUserName">Oxxx Nxxxxxxx</Data>-
<Data Name="TargetLogonId">0x0000000000015a3c</Data>-
</EventData>-</Event>-
```

Figure 20 Logoff date, time, and username (extract from Security Event log)

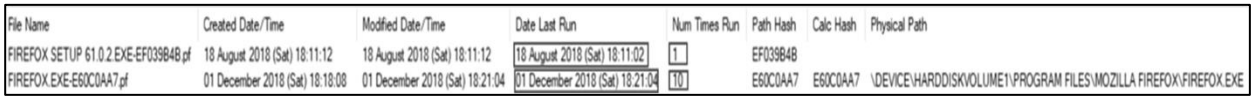
The artifacts presented in this section provide additional means of contextualizing the URL artifacts discussed in Section 5.1.1. This information shows that the browser session activity, as discussed in Section 5.1.2.2 is within the time frame of the user's last logon and logoff on the system.

#### 5.1.2.5 Prefetch

Prefetch is used by Windows for memory management, speeding up the boot process and application startup process. Prefetch files can be found in %SYSTEMROOT%\Prefetch and have a .pf extension.

Prefetch helps an investigator determine when an application was installed when it was last run, and the number of times the application was run. In the context of this research, Prefetch provides information related to Firefox; thus, corroborating other artifacts recovered previously. Figure 21 shows the prefetch files for when Firefox was

installed, the last time it was launched and the number of times it had been launched since installation.



| File Name | Created Date/Time | Modified Date/Time | Date Last Run | Num Times Run | Path Hash | Calc Hash | Physical Path |
|---|---|---|---|---|---|---|---|
| FIREFOX SETUP 61.0.2.EXE-EF039B4B.pf | 18 August 2018 (Sat) 18:11:12 | 18 August 2018 (Sat) 18:11:12 | 18 August 2018 (Sat) 18:11:02 | 1 | EF039B4B | | |
| FIREFOX.EXE-E60C0AA7.pf | 01 December 2018 (Sat) 18:18:08 | 01 December 2018 (Sat) 18:21:04 | 01 December 2018 (Sat) 18:21:04 | 10 | E60C0AA7 | E60C0AA7 | \DEVICE\HARDDISKVOLUME1\PROGRAM FILES\MOZILLA FIREFOX\FIREFOX.EXE |

Figure 21 Firefox last run date and run count

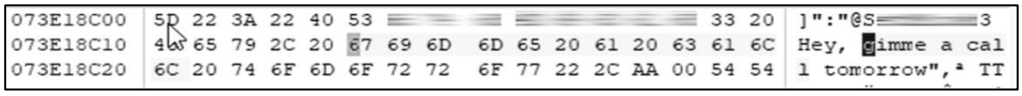
#### 5.1.2.6 Keyword Search

As the last task of the second stage of the model, a keyword search was used to recover artifacts resident in other parts of the disk such as unallocated space, slack space, and pagefile. The simultaneous search feature in WinHex was used to search across a variety of character encodings. A keyword search is useful in the identification and recovery of outlier artifacts that may be resident in the unstructured part of a disk where they are not readily visible or accessible when viewed in a digital forensics tool.

It is important to use search terms or strings that would reduce the number of false positives returned. This may involve using some of the features extracted from the URLs or keys from JSON data discussed in Sections 5.1.1 and 5.1.2.2. Examples of the search terms used include but is not limited to:

- "followed_by":
- follower_requests
- profile_image_url
- is_following_advertiser

Figure 22 shows an example of the results returned by the keyword search includes the tweet sent by the user.



```
073E18C00  5D 22 3A 22 40 53 [redacted]  33 20  ]":"@S[redacted]3
073E18C10  48 65 79 2C 20 67 69 6D  6D 65 20 61 20 63 61 6C  Hey, gimme a cal
073E18C20  6C 20 74 6F 6D 6F 72 72  6F 77 22 2C AA 00 54 54  l tomorrow",ª TT
```

Figure 22 Tweet Fragment from unallocated space

Keyword searches also highlight artifacts that may be of interest. For instance, Figure 23 shows an example of a Google Analytics (GA) URL indicating user login activity in the payload data. Table 9 breaks down the login URL. Additional parameters from the GA URL also allude to cookies set during the session (see Figure 24).

```
6934584320                68 74 74 70  73 3A 2F 2F 77 77 77 2E       https://www.
6934584336   67 6F 6F 67 6C 65 2D 61  6E 61 6C 79 74 69 63 73   google-analytics
6934584352   2E 63 6F 6D 2F 63 6F 6C  6C 65 63 74 3F 76 3D 31   .com/collect?v=1
6934584368   26 5F 76 3D 6A 37 32 26  61 69 70 3D 31 26 61 3D   &_v=j72&aip=1&a=
6934584384   32 39 34 37 38 31 38 35  33 26 74 3D 70 61 67 65   294781853&t=page
6934584400   76 69 65 77 26 5F 73 3D  32 26 64 6C 3D 68 74 74   view&_s=2&dl=htt
6934584416   70 73 25 33 41 25 32 46  25 32 46 74 77 69 74 74   ps%3A%2F%2Ftwitt
6934584432   65 72 2E 63 6F 6D 25 32  46 73 65 74 74 69 6E 67   er.com%2Fsetting
6934584448   73 25 32 46 73 61 66 65  74 79 26 64 72 3D 68 74   s%2Fsafety&dr=ht
6934584464   74 70 73 25 33 41 25 32  46 25 32 46 74 77 69 74   tps%3A%2F%2Ftwit
6934584480   74 65 72 2E 63 6F 6D 25  32 46 73 65 74 74 69 6E   ter.com%2Fsettin
6934584496   67 73 25 32 46 73 61 66  65 74 79 26 64 70 3D 25   gs%2Fsafety&dp=%
6934584512   32 46 75 73 65 72 25 32  46 68 6F 6D 65 25 32 46   2Fuser%2Fhome%2F
6934584528   68 6F 6D 65 26 75 6C 3D  65 6E 2D 67 62 26 64 65   home&ul=en-gb&de
6934584544   3D 55 54 46 2D 38 26 64  74 3D 54 77 69 74 74 65   =UTF-8&dt=Twitte
6934584560   72 26 73 64 3D 32 34 2D  62 69 74 26 73 72 3D 31   r&sd=24-bit&sr=1
6934584576   30 32 34 78 37 36 38 26  76 70 3D 31 30 30 37 78   024x768&vp=1007x
6934584592   36 35 34 26 6A 65 3D 30  26 5F 75 3D 53 41 43 41   654&je=0&_u=SACA
6934584608   41 51 41 42 7E 26 6A 69  64 3D 26 67 6A 69 64 3D   AQAB~&jid=&gjid=
6934584624   26 63 69 64 3D 38 33 37  39 39 38 39 36 35 2E 31   &cid=837998965.1
6934584640   35 34 33 36 38 38 35 31  35 26 74 69 64 3D 55 41   543688515&tid=UA
6934584656   2D 33 30 37 37 35 2D 36  26 5F 67 69 64 3D 31 33   -30775-6&_gid=13
6934584672   37 32 33 37 30 37 31 36  2E 31 35 34 33 36 38 38   72370716.1543688
6934584688   35 31 35 26 7A 3D 34 37  33 37 33 33 30 30 35 00   515&z=473733005
```

```
https://www.google-analytics.com/collect?v=1&_v=j72&aip=1&a=1803425654
&t=pageview
&_s=1
&dl=https%3A%2F%2Ftwitter.com%2F
&dr=https%3A%2F%2Ftwitter.com%2Flogin
&dp=%2Fuser%2Fhome%2Fhome
&ul=en-gb
&de=UTF-8
&dt=Twitter
&sd=24-bit
&sr=1024x768
&vp=1007x617
&je=0
&_u=QACAAQAB~
&jid=
&gjid=
&cid=837998965.1543688515
&tid=UA-30775-1ca8eb7406
&_gid=1372370716.1543688515
&z=888718953
```

Figure 23 Twitter login activity captured in google analytics URL

Other GA URLs returned from the keyword search indicates user access to "settings/account", "settings/safety", "safety/security" etc. on Twitter. When investigating social network activity, GA URLs, if on disk, may be useful in understanding user activity.

Table 9 Google analytics URL parameter breakdown (Google Developers, 2018)

| Parameter | Value | Description |
|---|---|---|
| t | pageview | This is the 'Hit' type. Permitted values for this parameter are 'pageview', 'screenview', 'event', 'transaction', 'item', 'social', 'exception', 'timing'. |
| _s | 1 | Hit sequence. The value increments by 1 with each pageview hit. |
| dl | https://twitter.com/ | Document location URL: This parameter sends a resource (or document) location. |
| dr | https://twitter.com/login | Document referrer: the format for the value for this parameter is a URL (and specifies the referral source of traffic). |
| dp | /user/home/home | Document path (i.e. resource path) specifies the 'path' portion of the URL. |
| ul | en-gb | User language |
| de | UTF-8 | This specifies the character set used in encoding the page / resource (twitter). |
| dt | Twitter | Document title. In this instance, "Twitter" is the page title. |

| baseDomain | name | value | host | path |
|---|---|---|---|---|
| twitter.com | dnt | 1 | .twitter.com | /settings/safety/ |
| twitter.com | fm | 0 | .twitter.com | / |
| twitter.com | _gat | 1 | .twitter.com | / |
| twitter.com | _ga | GA1.2.837998965.1543688515 | .twitter.com | / |
| twitter.com | _gid | GA1.2.1372370716.1543688515 | .twitter.com | / |
| twitter.com | personalization_id | "v1_hu[redacted]WyA==" | .twitter.com | / |
| twitter.com | guest_id | v1%3A154368[redacted]89493 | .twitter.com | / |
| twitter.com | ct0 | 61ddd5fe191ebcee41d753de50f1499d | .twitter.com | / |
| twitter.com | eu_cn | 1 | .twitter.com | / |
| twitter.com | ads_prefs | "HBISAAA=" | .twitter.com | / |
| twitter.com | kdt | MvecUTA[redacted]SO0q1mQ372 | .twitter.com | / |
| twitter.com | remember_checked_on | 1 | .twitter.com | / |
| twitter.com | twid | "u=1031[redacted]496" | .twitter.com | / |
| twitter.com | auth_token | 5a[redacted] | .twitter.com | / |
| twitter.com | csrf_same_site_set | 1 | .twitter.com | / |
| twitter.com | csrf_same_site | 1 | .twitter.com | / |
| twitter.com | dnt | 1 | .twitter.com | / |

Figure 24 Cookies.sqlite shows cookie info found in google analytics URL

Recovering and correctly interpreting artifacts such as the ones discussed in this section will enable the investigator/analyst to explain the important aspects of the recovered URL features in the context of a user's social relationships and activity. For example, explaining:

i. a user's connection to a social networking site e.g., account set up and credentials

ii. a user's social relationships, e.g., if the user is following or being followed by another user
iii. whether Twitter IDs found are as a result of direct contact, sponsored content or background processes

## 6. CONCLUSION AND FUTURE WORK

This paper has proposed a two-stage model for investigating social network activity. It has shown that a user's social networking activity can be inferred based on a range of artifacts, and it is possible based on these artifacts recovered, to identify and prioritize evidence that is pertinent to a case.

Although the syntax (protocol://hostname/path) is constant, some parts of URLs may be subject to change in structure due to improvements/changes implemented by the service provider. Such changes may include the implementation of shortened URLs or changing the location of resources on the website, e.g., moving user photos from /home to /home/profile.

This model is currently focused on manual analysis with the help of digital forensics tools. Further work is required to improve this model by automating each stage of the evidence recovery process and testing its generalizability and applicability across various browsers and operating systems.

## REFERENCES


[1] ACPO. (2012). *Good Practice Guide for Computer-Based Electronic Evidence v5.* Association of Chief Police Officers of England, Wales & Northern Ireland.

[2] Agency. (2015). *Five internet trolls a day convicted in UK as figures show ten-fold increase.* Retrieved July 2016, from The Telegraph: https://www.telegraph.co.uk/news/uknews/law-and-order/11627180/Five-internet-trolls-a-day-convicted-in-UK-as-figures-show-ten-fold-increase.html

[3] Arshad, H., Jantan, A., & Omolara, E. (2019). Evidence collection and forensics on social networks: Research challenges and directions. *Digital Investigation, 28*, 126-138.

[4] BBC News. (2010). *Facebook murderer to serve at least 35 years*. Retrieved August 2010, from BBC News: http://news.bbc.co.uk/1/hi/england/wear/8555221.stm

[5] BBC News. (2012). *Huge rise in social media "crimes"*. Retrieved May 2013, from BBC News: https://www.bbc.co.uk/news/uk-20851797

[6] Bello, M. D. (2013). *Twitter: The new face of crime*. Retrieved April 2017, from USA Today: http://www.usatoday.com/story/news/nation/2013/09/29/twitter-crime-dark-side/2875745/

[7] Berners-Lee, T., Masinter, L., & McCahill, M. (1994). *RFC 1738 - Uniform Resource Locators (URL)*. Retrieved November 2010, from IETF Network Working Group - Request for Comments: http://www.ietf.org/rfc/rfc1738.txt

[8] Bowcott, O., Carter, H., & Clifton, H. (2011). *Facebook riot calls earn men four-year jail terms amid sentencing outcry*. Retrieved September 2012, from The Guardian: https://www.theguardian.com/uk/2011/aug/16/facebook-riot-calls-men-jailed

[9] Cambridge University Press. (2019). Cambridge Dictionary. Cambridge University Press.

[10] Carvey, H. (2018). *RegRipper*. Retrieved November 2019, from GitHub: https://github.com/keydet89/RegRipper2.8

[11] Case, A., & Marziale, L. (n.d.). RegistryDecoder.

[12] Casey, E. (2002). Error, Uncertainty, and Loss in Digital Evidence. *International Journal of Digital Evidence, Summer*(1).

[13] Casey, E. (2005). *Computer Crime and Digital Evidence, in Encyclopedia of Forensic and Legal Medicine.* Oxford: Elsevier.

[14] Casey, E. (2011). *Foundations of Digital Forensics, in Digital Evidence and Computer Crime. Forensic Science, Computers and the Internet.* Elsevier Academic Press.

[15] Chisum, W. J., & Turvey, B. E. (2000). Evidence Dynamics: Locard's Exchange Principle & Crime Reconstruction. *Journal of Behavioural Science, 1*, 1-10.


© 2020 JDFSL 

[16] Chisum, W. J., & Turvey, B. E. (2007). A History of Crime Reconstruction. In *Crime Reconstruction* (Vol. Chapter 1, pp. 1-36). Elsevier Academic Press.

[17] Crown Prosecution Service (CPS). (2018). *Guidelines on prosecuting cases involving communications sent via social media*. Retrieved March 2019, from cps.gov.uk: https://www.cps.gov.uk/legal-guidance/social-media-guidelines-prosecuting-cases-involving-communications-sent-social-media

[18] Cusack, B., & Son, J. (2012). Evidence Examination Tools for Social Networks. *10th Australian Digitital Forensics Conference* (pp. 33-40). SRI Security Research Institute, Edith Cowan University, Perth, Western Australia.

[19] Das, D., Medhi, S. P., & Shaw, U. (2016). Social Network Forensics: Survey and Challenges. *International Journal of Computer Science and Information Security (IJCSIS), 14*, 310–316.

[20] DB4S Project. (n.d.). *DB Browser for SQLite*. Retrieved February 2019, from http://sqlitebrowser.org/

[21] Garfinkel, S. L. (2006). Forensic Feature Extraction and Cross-Drive Analysis. *Digital Investigation, 3S*, 71-81.

[22] Garfinkel, S. L. (2013). Digital media triage with bulk data analysis and bulk_extractor. *Computer Security, 32*, 56-72.

[23] Haroon, S., & Carter, H. (2010). *Facebook security measures criticised after Ashleigh Hall murder*. Retrieved August 2010, from The Guardian: http://www.theguardian.com/uk/2010/mar/09/ukcrime-facebook

[24] House of Lords Select Committee on Communications. (2014). *CHAPTER 2: SOCIAL MEDIA AND THE LAW*. Retrieved March 2019, from publications.parliament.uk: https://publications.parliament.uk/pa/ld201415/ldselect/ldcomuni/37/3704.htm

[25] Huber, M., Mulazzani, M., Leithner, S., Schrittwieser, M., Wondracek, G., & Weippl, E. (2011). Social Snapshots: Digital Forensics for Online Social Networks. *Proceedings of the 27th Annual Computer Security Applications Conference* (pp. 113–122). Orlando, Florida: ACM.

[26] Jang, Y. J., & Kwak, J. (2015). Digital forensics investigation methodology applicable for social network services. *Multimedia Tools and Applications, 74*, 5029–5040.

[27] Jonsson, P. (2011). *"Flash robs": How Twitter is being twisted for criminal gain [VIDEO]*. Retrieved May 2018, from Christian Science Monitor: http://www.csmonitor.com/USA/2011/0803/Flash-robs-How-Twitter-is-being-twisted-for-criminal-gain-VIDEO.

[28] Keyvanpour, M., Moradi, M., & Hasanzadeh, F. (2014). Digital forensics 2.0: A review on social networks forensics. *Studies in Computational Intelligence , 555*, 17-46.

[29] Mabuto, E. K., & Venter, H. S. (2012). User-generated digital forensic evidence in graphic design applications. *2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec 2012)* (pp. 195–200). IEEE.

[30] McGuire, M. (2019). *Into The Web of Profit: Social Media Platforms and the Cybercrime Economy*. Retrieved October 2019, from Bromium: https://www.bromium.com/wp-content/uploads/2019/02/Bromium-Web-of-Profit-Social-Platforms-Infographic.pdf

[31] McKemmish, R. (2008). When is Digital Evidence Forensically Sound? *IFIP - The Internation Federation for Information Processing, 285*(1), 3-15.

[32] Moore, K. (2014). *Social media 'at least half' of calls passed to front-line police*. Retrieved December 2015, from BBC News: https://www.bbc.co.uk/news/uk-27949674

[33] Murr, M. (2007). *The admissibility vs. weight of digital evidence | Forensic Computing.* Retrieved November 2010, from Forensic Computing: https://forensicblog.org/the-admissibility-vs-weight-of-digital-evidence/

[34] NirSoft. (2018). *FullEventLogView*. Retrieved December 2018, from https://www.nirsoft.net/utils/full_event_log_view.html

[35] NirSoft. (2018). *MZCacheView*. Retrieved December 2018, from https://www.nirsoft.net/utils/mozilla_cache_viewer.html

[36] Oh, J., Lee, S., & Lee, S. (2011). Advanced evidence collection and analysis of web

browser activity. *Digital Investigation, 8*(Supplement), S62-S70.

[37] Osborne, B. (2010). *Twitter sees more active users, but also attracts more criminal activity* . Retrieved from GEEK.COM: http://www.geek.com/news/twitter-sees-more-active-users-but-also-attracts-more-criminal-activity-1130461/

[38] Powell, A., & Haynes, C. (2019). Social Media Data in Digital Forensics Investigations. *Digital Forensic Education*, 281 - 303.

[39] Press Association. (2014). *Peter Nunn jailed for abusive tweets to MP Stella Creasy*. Retrieved January 2015, from The Guardian: https://www.theguardian.com/uk-news/2014/sep/29/peter-nunn-jailed-abusive-tweets-mp-stella-creasy

[40] Python.org. (2019). *Python 3*. Retrieved July 2019, from https://www.python.org/downloads/

[41] Rankin, B. (2012). *Send in the "Twitter squad": Police forces may need dedicated to cope with rising social media crime*. Retrieved January 2014, from Mirror UK: http://www.mirror.co.uk/news/technology-science/technology/rocketing-crime-complaints-involving-social-1507527

[42] Richards, J. (2007). *Sex offenders can use social sites, say police*. Retrieved November 2010, from Times Online: https://www.thetimes.co.uk/article/sex-offenders-can-use-social-sites-say-police-nfvb6l9k6cb

[43] Sommer, P. (1999). Intrusion detection systems as evidence. *Computer Networks, 31*, 2477-2487.

[44] Taylor, M., Haggerty, J., Gresty, D., Almond, P., & Berry, T. (2014). Forensic investigation of social networking applications. *Digital Investigation, 11*, 9-16.

[45] Telerik. (2018). *Fiddler - Free Web Debugging Proxy*. Retrieved June 2018, from Telerik: https://www.telerik.com/fiddler

[46] Ultimate IT Security. (2014). *Windows Security Log Encyclopedia*. Retrieved January 2014, from Ultimate Windows Security: https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx

[47] Woan, M. (2013). *PrefetchForensics*. (PrefetchForensics) Retrieved June 2013, from GitHub: https://github.com/woanware/woanware.github.io/blob/master/downloads/PrefetchForensics.v.1.0.4.zip

[48] Wood, C. (2018). *WhatsApp photo drug dealer caught by "groundbreaking" work*. Retrieved April 2018, from BBC News: https://www.bbc.co.uk/news/uk-wales-43711477

[49] X-Ways Software Technology AG. (2018). *WinHex: Computer Forensics & Data Recovery Software, Hex Editor & Disk Editor*. Retrieved May 2018, from X-Ways: https://www.x-ways.net/winhex/

[50] Zainudin, N. M., Merabti, M., & Llewellyn-Jones, D. (2011). Online social networks as supporting evidence: A digital forensic investigation model and its application design. *International Conference on Research Innovation in Information Systems (ICRIIS'11)*. IEEE.



https://dspace.org/

# A two-stage model for social network investigations in digital forensics

**David, Anne; Morris, Sarah; Appleby-Thomas, Gareth J.**

**2020-08-20**
Association of Digital Forensics, Security and Law
https://commons.erau.edu/jdfsl/vol15/iss2/1
David A, Morris S, Appleby-Thomas G. (2020) A two-stage model for social network investigations in digital forensics. Journal of Digital Forensics, Security and Law, Volume 15, August 2020, Article number 1

Downloaded from a DSpace repository. https://dspace.org/